# A Multimodal Dataset for Object Model Learning from Natural Human-Robot Interaction

Pablo Azagra, Florian Golemo, Yoan Mollard, Manuel Lopes, Javier C Civera, Ana C Murillo

# A Multimodal Dataset for Object Model Learning from Natural Human-Robot Interaction

Pablo Azagra[1], Florian Golemo[2], Yoan Mollard[2], Manuel Lopes[3], Javier Civera[1], Ana C. Murillo[1]

***Abstract*— Learning object models in the wild from natural human interactions is an essential ability for robots to perform general tasks. In this paper we present a robocentric multimodal dataset addressing this key challenge. Our dataset focuses on interactions where the user teaches new objects to the robot in various ways. It contains synchronized recordings of visual (3 cameras) and audio data which provide a challenging evaluation framework for different tasks.**

**Additionally, we present an end-to-end system that learns object models using object patches extracted from the recorded natural interactions. Our proposed pipeline follows these steps: (a) recognizing the interaction type, (b) detecting the object that the interaction is focusing on, and (c) learning the models from the extracted data. Our main contribution lies in the steps towards identifying the target object patches of the images. We demonstrate the advantages of combining language and visual features for the interaction recognition and use multiple views to improve the object modelling.**

**Our experimental results show that our dataset is challenging due to occlusions and domain change with respect to typical object learning frameworks. The performance of common out-of-the-box classifiers trained on our data is low. We demonstrate that our algorithm outperforms such baselines.**

## I. Introduction

One of the key challenges in service robotics is to achieve an intuitive Human-Robot Interaction (HRI), that feels natural to the user. To achieve this, it is essential that the robot learns models in a realistic environment adapted to a particular domain. These models should include objects, scenes, affordances, and capabilities which, in addition, might change over time.

In this work we address the relevant and challenging scenario of a robot learning new object models by interacting with humans in a natural manner. Object models learned from general datasets miss the subtle details of the particular scenes the robot works in. For example a soda can from a specific brand might look the same everywhere, but the appearances of kitchen utensils may vary a lot. In a home deployment scenario existing objects can be modified and new unknown ones can appear. Finally, sensory noise, clutter, and illumination conditions might also change within a domain and cause standard classifiers to fail.

Our contribution along this line of research is twofold: Firstly, we release a partially annotated dataset for object modeling from natural human-robot interactions. Our dataset features robocentric multimodal data from several users and objects, which we found to be novel in this field. Secondly, we propose a full pipeline for acquiring object models from the recorded data (see Fig. 1 for an overview). To our knowledge, our proposal is the first one addressing such a multimodal and interactive setting. Our pipeline is designed to address the additional challenges (everyday object segmentation and recognition problems) posed by this natural HRI setup.

Our experimental results underline the challenges of the setting we propose. We need to understand which object the user refers to from the ones available in the scene. Therefore we propose a way to guide different strategies for target object segmentation thanks to a simple initial interaction recognition. Standard object recognition classifiers trained on state-of-the-art object recognition databases exhibit a low performance on our dataset. However, the recognition rate improves when these methods are trained on annotated data from our dataset. This confirms that our data is in a significantly different segment of the domain due to the particularly natural setting of HRI. We evaluate our pipeline and set an initial baseline, presenting promising results about the use of multimodal data to enable learning from noisy object patches.

## II. Related work

There are many works in the literature in which the robot interacts directly with the objects in a scene to learn new models. For example, Collet et al. [3] created a 3D model of the objects in the scene that a robotic hand has to grasp. Kenney et al. [7] proposed to improve object segmentation in cluttered scenarios by manipulating the objects. Additionally there are multiple works which use robotic hands to interact with objects in the scene. For example Iravani et al. [6] proposed a system where the robot manipulates the objects presented in front of the camera until the model is learned. Krainin et al. [8] proposed to use a robotic hand to grasp the object and rotate it to obtain different views. Sinapov et al. [16] used the robotic hands to interact with plastic jars and obtain multimodal information to learn the content of the jars. These approaches typically need prior information to be able to grasp the objects. Our approach is complementary to these works and focuses on scenarios that require human interaction, e.g. if the object is out the robot's reach the affordances are completely unknown or the grasping capabilities are limited.

For any given robotic platform intended to act in the real world it is necessary to obtain object models. In this sense the approach of Pasquale et al. [13] is very similar to ours. The authors created a dataset and used CNN-based

[1] DIIS-I3A. Universidad de Zaragoza, Spain.
[2] INRIA Bordeaux Sud-Ouest, France.
[3] INESC-ID, Instituto Superior Técnico, Univ. de Lisboa, Portugal


Fig. 1: Object model learning from human-robot interaction. In the proposed pipeline the user interacts with the robot for it to learn new object models. We capture audio and video from this interaction. The data is processed to recognize the interaction type which guides the segmentation of the target object patches in the images. These patches are used to learn new object models.

features and SVM classification for visual recognition. Their training data consists of robocentric images, where a human presents an object in front of the iCub [11]. We improve on these efforts by focusing our dataset and algorithm on realistic multimodality and multiple interaction types. Where these previous approaches solely relied on images we present video, audio, and depth information that can be obtained with common hardware. Furthermore, we extended the interaction to several types which posed the additional challenge of interaction recognition. The dataset presented in Vatakis et al. [19] includes similar multimodal recordings, but it is focused on the psychological reaction of the users.

There are countless datasets for object recognition, e.g., [10], [17] among the ones containing *RGB-D* images for object recognition or [15] using only *RGB* images but containing a enormous variety of objects. However, most of these contain clean images of the objects in a studio, or high resolution pictures of objects in the wild. Whereas such datasets can always be used for offline learning we place our dataset as more realistic by capturing the noise and clutter that would be encountered in an interactive scenario.

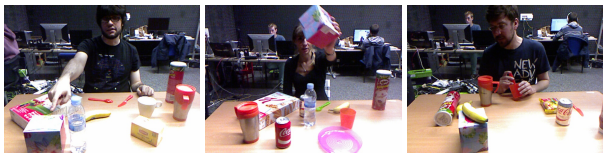
Our work also emphasizes that the point of view is crucial. For example, recognizing a pedestrian from a close-up view of a service robot is very different from performing the same task in the raw video from a distant wide-angle surveillance camera. Datasets like [18] or [5] capture human-robot interactions from an external point of view. In the case of mobile robots, using the onboard sensors is more practical than installing sensors everywhere the robot can go.

## III. Multimodal Human-Robot Interaction Dataset (MHRI)

Our "MHRI" dataset[1] captures the most common natural interactions to teach object classes to a robot, namely *Point*, *Show*, and *Speak*, from a robocentric perspective. Figure 2 shows an example for each considered interaction type (captured from the robot frontal camera):

- *Point*: the user points at an object on the table and announces its name.

[1]Available at `http://robots.unizar.es/IGLUdataset/`



(a) *Point* (b) *Show* (c) *Speak*

Fig. 2: Examples from the three interaction types in MHRI dataset. The user says, respectively, (a) "This is a box", while pointing at the box, (b) "This is a box", while holding the box, and (c) "The box is next to the chips and has a banana on top."

TABLE I: Summary of the dataset content

| | | |
|---|---|---|
| **Users** | 10 | |
| **Interaction Type** | 3 | *Point, Show, Speak* |
| **Interactions per User** | 30 | 10 of each type. 1 object per interaction. |
| **Object Pool** | 22 | *Apple, Banana, Big Mug, Bowl, Cereal Box, Coke, Diet Coke, Glass, Fork, Ketchup, Kleenex, Knife, Lemon, Lime, Mug, Noodles, Orange, Plate, Pringles, Spoon, Tea Box, Water Bottle* |

- *Show*: the user grabs an object, moves it closer to the robot, and utters its name.
- *Speak*: the user describes where a certain object is in relation to other objects.

Table I summarizes the contents of the dataset. It contains recordings from 10 users and each user performed 10 object interactions of each of the 3 types (*Point*, *Show*, *Speak*), for a total of 300 multimedia short clips. The aforementioned 10 objects per user were picked randomly out of a pool of 22 objects and used by that user for all their recordings. Figure 3 illustrates the different sensor modalities of the dataset for different users.

### A. Technical Information

The dataset contains 4 synchronized streams of data: 2 RGB-D video feeds, from frontal and top point of views, acquired with *Kinect v1* sensors), 1 RGB video feed from a $1280 \times 720$ HD camera, and 1 audio feed captured with a studio microphone. Table II shows the specific data formats

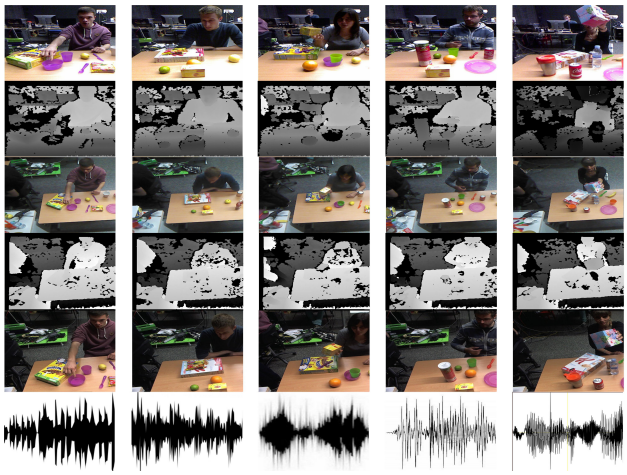


Fig. 3: Five examples (one user per column) from MHRI dataset. Each row displays a different sensor modality. From top to bottom: *Frontal*-RGB, *Frontal*-depth, *Top*-RGB, *Top*-depth, HD camera, and audio.

TABLE II: Dataset format specifications

| Device | Data | Format |
|---|---|---|
| RGB-D Cameras (Frontal & Top) | RGB frames | 640x480 JPEG |
| | Depth frames | 640x480 PNG |
| HD Camera | RGB frames | 1280x720 JPEG |
| Microphone | Audio file | 44.1kHz Stereo WAV |

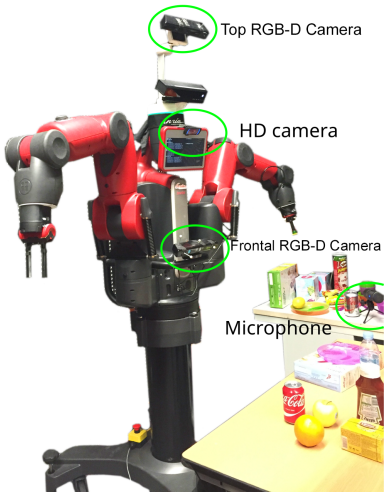



Fig. 4: Baxter robot used to acquire the dataset. The three cameras and the microphone locations are highlighted.

available and Fig. 4 shows the cameras placement in the Baxter robot used for the acquisition. The *Frontal* RGB-D camera is mounted on the robot chest to give a frontal view of the user and the table. The *Top* RGB-D camera is mounted at the highest point of the robot and has a holistic overview of the scene.

### B. Annotations

The dataset annotations include the list of the objects each user interacted with, the first uttered word (which is either "this", "that" or "the"), and the label of the object in question for each interaction. Additionally, each frame is timestamped (using ROS[2]) and labeled with the type of interaction (*Point*, *Show*, *Speak*).

## IV. Learning from Multimodal Interactions

This section presents the proposed pipeline for object learning by leveraging different data modalities that capture natural interactions. Our pipeline is composed of three modules, summarized in Fig. 1:

- **Interaction Recognition.** The extraction of candidate object patches depends on the interaction type (*Point*, *Show*, *Speak*), so an accurate identification of the interaction is crucial.
- **Target Object Detection.** For each interaction type we propose a specific algorithm to select the candidate image patches that are likely to contain the object.
- **Object Model Learning.** The candidate patches from the previous step are used as training examples for supervised learning (the class labels coming from the users' speech).

Our main contributions lie in the first two modules, as once we have extracted the target object patches we can use standard object model learning algorithms. We aim to demonstrate the benefits of the multiple data sources and the feasibility of learning from natural interaction. The next subsections detail each module.

### A. Multimodal Interaction Recognition

Classifying the type of interaction performed by a person using only visual data is considerably challenging. The work of [1] shows that the combination of language and vision can lead to a substantial improvement. Our interaction recognition module uses visual and language features in a nested SVM-based classification.

**Language Features:** We use a simple language feature consisting of the first word of the user's narration. In our dataset this word is either *this* or *that* for *Point* and *Show* interactions or any other word for the more descriptive *Speak* interaction. This feature is not discriminative enough to separate the three interaction classes, as we show in Fig. 5. It clearly separates *Speak* interactions, but cannot differentiate between *Point* and *Show*. Separating *Speak* is particularly valuable, as there are no specific visual patterns associated with this interaction.

**Visual Features:** Before computing the visual features, in order to focus on the user and table regions, we remove the background using two strategies: a standard background removal procedure, based on sliding-window average of all the previous frames, and a depth map based filter, where we remove all image pixels with a depth value over a threshold of $1.7m$ (based on the distance to the user and the table). We apply these two filters on the image and run a sliding-window filter (window size $100x100$ pixels, step of 10) over

[2] http://ros.org/

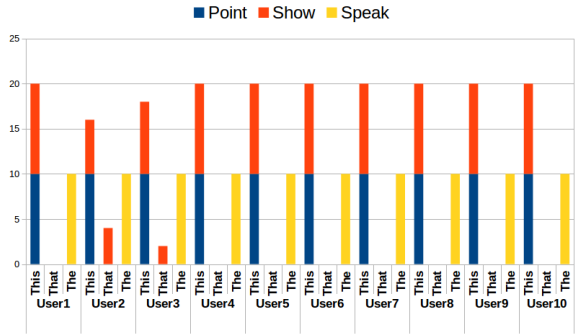



Fig. 5: Language feature occurrences in all recordings, per type of interaction and per user.

the masked image to reject windows where more than 30% of the pixels were removed by either one of these filters. Then we compute the visual descriptors on the accepted windows. We evaluate two different descriptors:

- *Color histograms* $HC = [H_r \ H_g \ H_b]$, with $H_i = \sum_{x,y} p_i(x,y) \bmod B$, where $p_i$ is pixel $i$ component value and $B$ the number of bins.
- *Histogram of Gradients* ($HOG$), as described in [4].

**Interaction Recognition:** We propose the following interaction recognition module, using the aforementioned language and visual features, based on two nested classifiers:

1) Binary discrimination between *Speak* videos and the other two types using the language features.
2) SVM classification into *hand* vs *no-hand* classes of sliding window-based patches, trained with random patches of the dataset and manually selected patches of hands. This step only uses the $HC$ descriptor due to its high efficiency and good performance at removing most of the *no-hand* patches.
3) SVM classification of resulting *hand* patches into *Point* or *Show* classes. Here we use both the $HC$ and the $HOG$ descriptors.
4) Assign a label, *Point* or *Show*, to each video according to the label obtained by the majority of its frames. All windows from each video are labeled as that action for the next step.

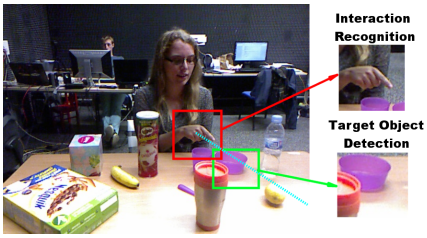
### B. Target Object Detection

The goal of this module is to extract image patches that are likely to contain the target object we want to model. Based on the results from the previous module, in particular *hand* patches from *Point* or *Show* classes, we propose two algorithms to segment the target object: one using the Frontal RGB-D camera only (named "Single-Camera") and another using the two RGB-D cameras ("Two-Cameras").

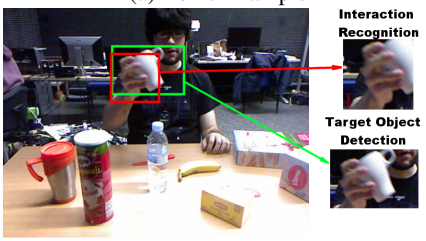
**Single-Camera algorithm:** We start by using SLIC [2] superpixels to segment the image and determine the object area efficiently. We propose different strategies (as detailed in Algorithm 1) to extract the target object patch depending on the interaction type:

```
Data: Video_RGB-D_Frontal,interaction,Hand_Pos
Result: Target Object Patch
if interaction == 'Point' then
    for each Frame in Video_RGB-D_Frontal do
        point_direction = get_pointing_direction(Hand_Pos);
        SuperPixels = SLIC(Frame);
        Sp_intersect = get_intersection(SuperPixels,point_direction);
        Patch = Patch_SP(Frame,Sp_intersect);
        add_patch(Patch);
    end
    Patches = get_patches(Patch);
    return Patches;
else
    for each Frame_F in Video_RGB-D_Frontal do
        Orientation= get_orientation(Frame,Hand_Pos);
        Patch = expand_Patch(Frame,Hand_Pos,Orientation);
        add_patch(Patch);
    end
    Patches = get_patches();
    return Patches;
end
```

**Algorithm 1: Single-Camera**. Target object detection using the Frontal RGB-D camera.




(a) Point Example




(b) Show Example

Fig. 6: Target Object Detection using **Single-Camera** algorithm. Two examples from different interaction types. The red box indicates the patch used to recognize the interaction type (hand patch) and the green box indicates the selected target object patch following the corresponding strategy. The dashed line in (a) is the estimated pointing direction.

- *Point interaction*: First, we estimate the pointing direction using the first-order moments of the hand superpixels. After that, we select the scene superpixel that intersects with the pointing vector. We extract the image patch that completely contains the intersecting superpixel. See Fig. 6(a) for an example.
- *Show interaction*: First, we estimate the hand superpixel orientation as the direction of its first-order moment. We assume that the object is aligned with the hand. Then we extract the image patch that contains the hand superpixel and the neighbouring superpixel following its orientation. See Fig. 6(b) for an example.

```
Data: Video_RGB-D_Frontal,Video_RGB-
      D_Top,interaction,Hand_Pos
Result: Target Object Patch
Candidates =
  Calculate_Candidates(Video_RGB-D_Frontal,Video_RGB-D_Top);
if interaction == 'Point' then
    for each Frame in Video_RGB-D_Frontal do
        point_direction = get_pointing_direction(Hand_Pos);
        for each Candidate in Candidates do
            dist,Inline = Intersect(Candidate,point_direction);
            if Inline && get_min_dist()> dist then
                set_candidate(Candidate,dist);
            end
        end
        add_vote(Candidate);
    end
    Winner = get_winner();
    Patches = extract_patches(Winner);
    return Patches;
else
    for each Frame_F in Video_RGB-D_Frontal do
        Orientation= get_orientation(Frame,Hand_Pos);
        Patch = expand_Patch(Frame,Hand_Pos,Orientation);
        add_patch(Patch);
    end
    Patches = get_patches();
    return Patches;
end
Function Calculate_Candidates(RGB-D_Frontal,
  RGB-D_Top):
    Homography = get_Homography();
    for First five Framesof RGB-D_Frontal, RGB-D_Top do
        Plane,θ= Calculate_plane(Frame_Top);
        Frame_Rotated= Rotate(Frame_Top,−θ);
        Area_Cropped = Extract_Plane_Crop(Frame_Top,Plane);
        Blobs = get_blobs(Area_Cropped);
        Projected_Blob= get_projection(Homography,Blobs);
        SuperPixels = SLIC(Frame_Frontal);
        Candidates =
          obtain_candidates(SuperPixels,Projected_Blob);
        return Candidates
    end
return
```

**Algorithm 2: Two-Cameras**. Target Object Detection using the two RGB-D cameras (Frontal and Top).

**Two-Cameras algorithm:** Our proposal leverages the two different points of view of the cameras in the MHRI dataset.

In the cases of *Show* interaction the object is easy to find in the Frontal camera because there is usually no occlusion. Here we are considering the more interesting case of *Point* interactions.

First, we search for an object candidate in the Top camera in the initial frames before the actual interaction has started. The Top camera allows us to get an accurate object pre-segmentation by subtracting the table plane. In this top view the objects are not affected by occlusions (they are in the frontal view). We have calibrated the table plane homography, so we can map approximately the objects from the Top to the Frontal camera. Once the candidates have been mapped, we calculate the closest object that intersects with the pointing direction in each frame. The object with the most votes within the video is the chosen one, and the image patches containing it are used as training samples. The details can be found in Algorithm 2. Fig 7 shows an illustrative example.

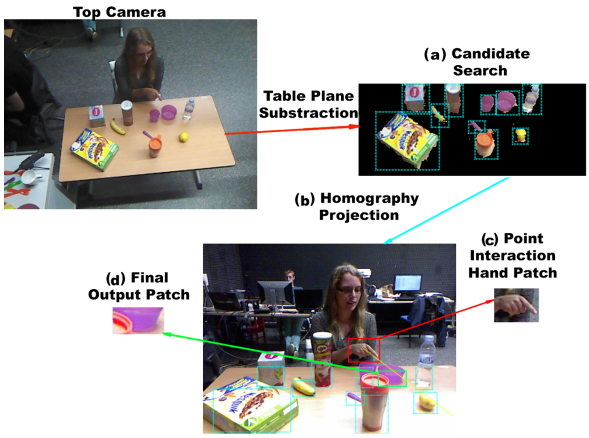



Fig. 7: Target Object Detection using **Two-Cameras**: (a) extract candidate objects in the Top camera, (b) project them on the Frontal camera, (c) find pointing direction from the hand patch in Frontal camera and (d) select object candidate that intersects with the pointing direction.

### C. Object Model Learning

In order to evaluate the proposed target object detection in the context of the presented dataset, we implemented a standard object model learning approach:

- ***Descriptors.*** We use *Color histograms* (*HC*) and *Bag of Words* (*BoW*), following [12]. The BoW model is built from ORB [14] extracted on images from the Washington dataset [10]. We used a standard k-means to build the vocabulary (with k=1000).
- ***Classifiers.*** We evaluate two classifiers: Support Vector Machines (**SVM**) and Nearest Neighbors (**NN**).

We disregarded naive Convolutional Neural Nets (CNN) as baseline because (a) the data given in our domain would not suffice to train a model from scratch and (b) transfer learning from a large object recognition dataset could also fail, as we are dealing with significant occlusions and imperfect segmentations.

## V. Experimental Results

This section presents several experiments to demonstrate the challenges of the proposed MHRI dataset. The following experiments validate our pipeline to recognize interaction types and automatically segment target object patches. Finally, it is also our aim to establish a baseline for the dataset.

In the following experiments, we consider four types of patches, all of them containing approximately one object:

- ***Washington Patches*** which contain correctly labeled objects from Washington dataset [10].
- ***Manual Patches*** which are manually cropped around objects from MHRI dataset images.
- ***Automatic Patches*** which are automatically obtained using our target object detection algorithm in the MHRI data.
- ***Inspected Patches*** which are a subset of the *Automatic Patches* that were visually inspected to verify that both patch and label are correct.

TABLE III: Interaction recognition accuracy.

| | **Point** | **Show** | **Speak** |
|---|---|---|---|
| **Point** | **72.85%** | 76.01% | 55.46% |
| **Show** | 12.36% | **12.88%** | 20.00% |
| **Speak** | 14.78% | 11.11% | **24.54%** |

(a) Vision-Only Classification

| **Point** | **Show** | **Speak** |
|---|---|---|
| **85.71%** | 22.03% | 0.00 |
| 14.29% | **77.97%** | 0.00 |
| 0.00% | 0.00% | **100.00%** |

(b) Multimodal Classification

TABLE IV: Target object patches in videos from each user.

| User: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | ALL |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *Point* videos | | | | | | | | | | | |
| #P | 68 | 62 | 75 | 31 | 42 | 32 | 55 | 38 | 65 | 60 | 53 (25) |
| #F | 130 | 91 | 99 | 64 | 62 | 72 | 77 | 78 | 81 | 72 | 82 (24) |
| *Show* videos | | | | | | | | | | | |
| #P | 133 | 110 | 125 | 97 | 124 | 100 | 136 | 149 | 133 | 123 | 123 (29) |
| #F | 183 | 135 | 136 | 114 | 136 | 112 | 148 | 170 | 141 | 132 | 141(28) |

#P: total number of object patches extracted

#F: total number of frames per video for this user

#ALL: average and standard deviation for all users and videos

Fig. 8 shows examples of the different types of patches. As *Inspected patches* is a subset of *Automatic patches* and their visual appearance is similar, examples of the former are excluded from the figure.

### A. Interaction Recognition

In order to demonstrate the benefits of multimodal data, we first classify the interaction type by using only visual data and SVM. Table III(a) shows the confusion matrix.

We augment the model with the speech modality using the first word of the user speech (*this/that/the*), as explained in Sec. IV. Table III(b) shows the confusion matrix obtained by this classifier, which improves the results for all classes, discriminating the *Speak* interaction and improving *Point* and *Show* from $72.85\%$ to $85.71\%$ and $12.88\%$ to $77.97\%$ respectively.

### B. Target Object Detection

Our aim in this section is to evaluate the quality of the *Automatic patches* obtained by our algorithm. Fig. 8 illustrates, qualitatively, the different visual appearance of the *Washington patches* and *Manual patches*. Notice how the *Automatic patches*, obtained from natural interactions, present several challenges for standard object learning methods. There is clutter around the target object and in most of the *Show* examples the hand significantly occludes the object. In many patches from *Point* interactions the target object is not centered and only partially visible. Table IV shows the average number of patches extracted with our approach for *Point* and *Show* clips.

As previously mentioned, we have evaluated deep learning features to model the target object patches. Unfortunately, the domain where object recognition models are usually trained, such as *ImageNet*, contains mostly clean images of complete objects. As Fig. 8c shows, the patches we extract contain mostly partial views of the objects, due to noisy segmentations and occlusions. Using existing CNNs to extract features in our patches did not improve our results. Fig. 9 shows a few classification examples of MHRI patches which contain object classes included in *AlexNet* model [9]. We can see that the CNN model fails to recognize the object if the patch does not contain the entire object.

TABLE V: Average accuracy (*Manual patches*).

| | **SVM** | | **NN** | |
|---|---|---|---|---|
| | **BOW** | **HC** | **BOW** | **HC** |
| **Manual** | 16.5-5.7-30.0 | 38.5-7.7-55.7(*) | 8-5.5-17.8 | 28.6-8.8-42.2 |

Accuracy-Std.Deviation-Best Experiment

* Confusion Matrix is shown in Fig. 10

### C. Object Model Learning

The following experiments are designed to evaluate the performance that learning algorithms can obtain in the MHRI dataset. As already mentioned, the challenges are many. The target object patches extracted automatically are noisy, have occlusions and are of low resolution in comparison to other datasets (see Fig. 8).

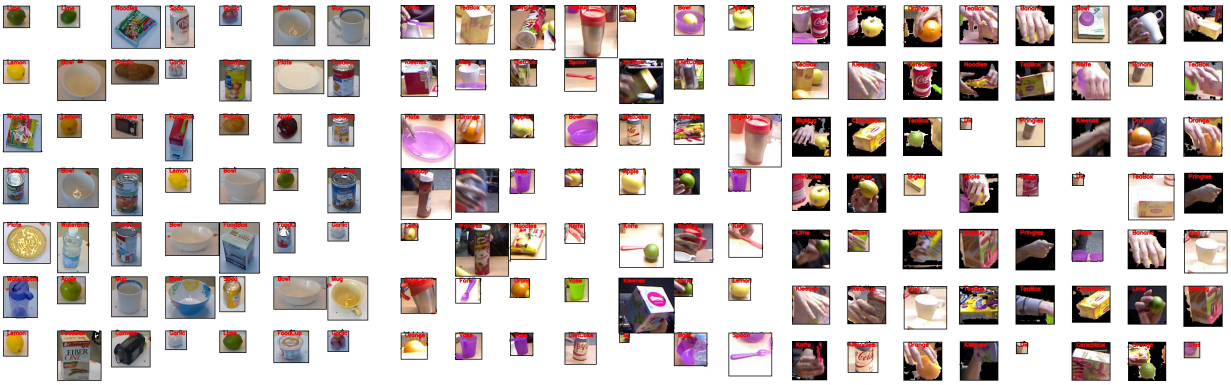
For the experiments in this section, we used the different types of image patches defined before (*Washington*, *Manual*, *Automatic* and *Inspected*) to train an object classifier. We do 10-fold cross-validation for the MHRI patches, each fold being all the images from one user. This ensures basic generalization over position, clutter, lighting and user bias in training and testing. The accuracy is averaged over the ten folds for the 22 objects in the MHRI dataset. Although each user only handles a subset of the objects, all the confusion matrices are of the same size, each of the 22 objects per row, following the same object order as Table I. The rows corresponding to unused objects are set to zero (displayed in black).

***Washington and Manual patches*:** Both sets of patches serve as baselines to evaluate the full object model pipeline. The *Washington patches* illustrate the domain change of our dataset. And we use the *Manual patches* to set an upper bound for the performance of the *Automatic patches*.

In a first experiment, we train a standard object recognition algorithm (BoW+SVM) using the *Washington patches* and evaluate its performance on *Manual patches* of the MHRI dataset. The average accuracy for the 12 classes that both datasets have in common is close to random (**8.4%**). This demonstrates that even when they share several objects the respective biases of the datasets cause naive approaches with pre-trained models to fail.

Our second experiment is to train several classifiers with *Manual patches*, and evaluate their performance in a test set from the same dataset. Table V shows the results. Notice that their best accuracy, **38.5%**, is considerably higher than the previous one trained on the *Washington patches*, confirming the dataset bias. Notice that SVM shows a better performance than NN, which is why we pick this classifier for the rest of the experiments.

Finally, observe that an accuracy of $38.5\%$ is low for supervised visual classification with manually annotated data. This result shows the challenging nature of our dataset and motivates its release.



(a) *Washington* patches (b) *Manual MHRI patches* (c) *Automatic MHRI patches*

Fig. 8: Examples of three types of object patches used in our experiments. Notice the increasing levels of clutter and segmentation noise.

banana rapeseed water-bottle milk-can bowl petri-dish

Fig. 9: Object labels for different sample patches using a pre-trained CNN (*AlexNet*). It often fails when patches contain only partial views of objects (banana and water bottle) or unexpected points of view (bowl).

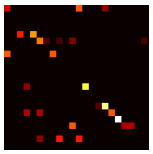


Fig. 10: **Confusion Matrix** for the best user in the experiment with *Manual patches* Lighter color is higher accuracy. Black rows correspond to unused objects.

TABLE VI: Accuracy (*Single-Camera vs Two-Cameras*, train with *Manual patches*, test with *Automatic patches*).

| | **Frontal Camera** | | **Two Cameras** | |
|---|---|---|---|---|
| | **BOW** | **HC** | **BOW** | **HC** |
| **Automatic** | 5.6-2.4-10.2 | 7.1-4.1-16.0 | 7.7-6.5-30.5 | 9.1-6.9-28.0(*) |

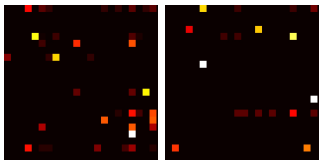
Accuracy-Std.Deviation-Best Experiment

* Confusion Matrix is shown in Fig. 11a

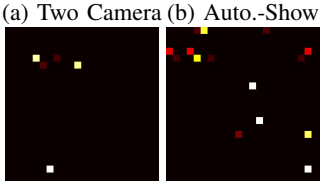
***Automatic patches*:** We present results for our full pipeline extracting image patches automatically using the approach in section IV.

*Single-Camera vs Two-Cameras:* In this experiment we show the benefit of using two cameras, as explained in section IV-B. Table VI shows the accuracy for our pipeline using one and two cameras. Notice that using two cameras improves the performance, both in average and best experiment accuracy. For the rest of the results in the paper we use the *Two-Cameras* algorithm.

***Inspected patches*:** The aim of the following experiments is to analyze the performance per-interaction type.



(a) Two Camera (b) Auto.-Show



(c) Insp.-Point (d) Insp.-HC

Fig. 11: **Confusion Matrix** for best user in the experiment with *Automatic patches*.

We use the *Manual patches* for training and the *Automatic patches* and *Inspected patches* for test. Table VII shows the accuracy results for this experiment. *Show* presents a higher accuracy (**9.6%**) than *Point* (**4.5%**) for *Automatic patches*. The patch extraction is more noisy for *Point*, due to the uncertainty associated with the pointing direction estimation. Notice however, that the performance of *Point* is much higher (**20.1%**) if we use *Inspected patches*. The reason is, if the pointing direction is accurately estimated, the *Point* patches are less affected by occlusion and hence the model learned is better. Observe also that, in general, the HC descriptor is better than the BoW one due to the low resolution of the patches and the occlusions.

TABLE VII: *Automatic patches* vs *Inspected patches* object recognition trained with manual patches.

| | **Show (75%)** | | **Point (25%)** | |
|---|---|---|---|---|
| | **BOW** | **HC** | **BOW** | **HC** |
| Automatic | 8.0-8.2-31.1 | 9.6-6.5-18.4(*) | 4.5-7.3-23.0 | 3.0-6.9-21.3 |
| Inspected | 7.6-10.4-35.1 | 8.6-6.8-19.2 | 8.4-8.0-27.3 | 20.1-21.7-66.6(**) |

Accuracy-Std.Deviation-Best Experiment

* Confusion Matrix is shown in Fig. 11b

** Confusion Matrix is shown in Fig. 11c

Finally, Table VIII shows the results using *Inspected*

TABLE VIII: Accuracy using *Inspected patches*. Confusion Matrix of best example in Fig. 11d

| User: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | ALL |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *Point* videos | | | | | | | | | | | |
| #P | 0 | 11 | 22 | 0 | 53 | 0 | 6 | 36 | 92 | 2 | 22 |
| #H(%) | 0 | 0 | 0 | 0 | 70 | 0 | 0 | 0 | 59 | 0 | 18 |
| *Show* videos | | | | | | | | | | | |
| #P | 47 | 100 | 111 | 47 | 88 | 76 | 114 | 73 | 9 | 116 | 78 |
| #H(%) | 9 | 43 | 35 | 49 | 10 | 45 | 12 | 19 | 11 | 12 | 24 |

#P: Average number of object patches accepted after inspection ;
#H: Accuracy classifying them; #ALL: Average for all users and videos

*patches* per user. We would like to highlight the high variability between users. This variability is promising; as the average accuracy is reasonably high for certain users, we believe the future research should focus on the challenging cases.

## VI. Conclusions

In this paper we have presented an annotated multimodal dataset for object model learning from human-robot interaction. The most remarkable features of the dataset are, first, the synchronized recording of multimodal data (two *RGB-D* cameras, one high resolution *RGB* camera and audio data), and second, the recording of natural interactions for humans (*Point*, *Show* and *Speak*) when teaching new object models. The dataset has a robocentric perspective.

As a second contribution, we have presented a first approach to object learning using multimodal data from natural HRI. Such approach is the initial baseline for the dataset, showing the feasibility and challenges of object learning from natural interactions. Our proposed algorithm also serves to demonstrate how the interaction classification benefits from the use of multimodal data (language and vision). The interaction recognition is a critical step, as the training data has to be extracted differently depending on the particular interaction. We have proposed a target object detection method to extract patches containing the objects, evaluated its performance, and shown its challenges. Finally, we have evaluated the full pipeline against manually extracted data. Our main conclusions are the following; First, the domain change is critical, and hence it is impractical to use data from other object datasets in a naive manner. Second, although our approach shows a reasonable performance, there are still considerable challenges in the target object detection and model learning, justifying the relevance of the presented dataset.

In future lines of work, we plan to improve the detection of the direction of the hand, develop new features that take advantages of the depth information, study the use of CNNs for object proposal and create an incremental learning system to discard the noisy or incorrectly labeled patches.

## ACKNOWLEDGMENT

This research has been partially funded by the European Union (CHIST-ERA IGLU, PCIN-2015-122, EU FP7-ICT project 3rdHand 610878), the Spanish Government (projects DPI2015-67275, DPI2015-65962-R, DPI2015-69376-R), the Aragon regional government (Grupo DGA T04-FSE), the University of Zaragoza (JIUZ-2015-TEC-03), and the Fundação para a Ciência e a Tecnologia (FCT) UID/CEC/50021/2013.

## References

[1] L. N. Abdullah and S. A. M. Noah. Integrating audio visual data for human action detection. In *Int. Conf. Computer Graphics, Imaging and Visualisation*, pages 242–246. IEEE, 2008.

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *Trans. on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012.

[3] A. Collet, D. Berenson, S. S. Srinivasa, and D. Ferguson. Object recognition and full pose registration from a single image for robotic manipulation. In *IEEE Int. Conf. on Robotics and Automation*, pages 48–55, May 2009.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition*, volume 1, pages 886–893. IEEE, 2005.

[5] W. Gong, J. Gonzàlez, J. M. R. S. Tavares, and F. X. Roca. *A New Image Dataset on Human Interactions*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[6] P. Iravani, P. Hall, D. Beale, C. Charron, and Y. Hicks. Visual object classification by robots, using on-line, self-supervised learning. In *IEEE Int. Conf. on Computer Vision Workshops*, pages 1092–1099, 2011.

[7] J. Kenney, T. Buckley, and O. Brock. Interactive segmentation for manipulation in unstructured environments. In *IEEE Int. Conf. on Robotics and Automation*, pages 1377–1382, 2009.

[8] M. Krainin, B. Curless, and D. Fox. Autonomous generation of complete 3D object models using next best view manipulation planning. In *IEEE Int. Conf. on Robotics and Automation*, pages 5031–5037, 2011.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[10] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view RGB-D object dataset, May 2011.

[11] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori. The iCub humanoid robot: an open platform for research in embodied cognition. In *Proceedings of the 8th workshop on performance metrics for intelligent systems*, pages 50–56. ACM, 2008.

[12] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 2161–2168. Ieee, 2006.

[13] G. Pasquale, C. Ciliberto, F. Odone, L. Rosasco, L. Natale, and I. dei Sistemi. Teaching iCub to recognize objects using deep convolutional neural networks. *Proc. Work. Mach. Learning Interactive Syst*, pages 21–25, 2015.

[14] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571. IEEE, 2011.

[15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[16] J. Sinapov, C. Schenck, and A. Stoytchev. Learning relational object categories using behavioral exploration and multimodal perception. In *IEEE Int. Conf. on Robotics and Automation*, pages 5691–5698, 2014.

[17] A. Singh, J. Sha, K. S. Narayan, T. Achim, and P. Abbeel. Bigbird: A large-scale 3d database of object instances. In *IEEE Int. Conf. on Robotics and Automation*, pages 509–516, 2014.

[18] J. Sung, C. Ponce, B. Selman, and A. Saxena. Human activity detection from RGBD images. *plan, activity, and intent recognition*, 64, 2011.

[19] A. Vatakis and K. Pastra. A multimodal dataset of spontaneous speech and movement production on object affordances. *Scientific Data*, Jan 2016.